Etienne Martineau <etmartin101@gmail.com>

FW: Guest VM running 25% slower than host ?

1 message

Etienne Martineau (etmartin) <etmartin@cisco.com> Tue, Jun 11, 2013 at 10:49 AM
To: "etmartin101@gmail.com" <etmartin101@gmail.com>

From: Etienne Martineau (etmartin)
Sent: Thursday, June 06, 2013 2:31 PM
To: Venkat Garigipati (venkatg); Pankaj Bhagra (bhagra); Alec Hothan (ahothan); bajor(mailer list); Ajay Patel (ajaypat); Rajesh Ranga (raranga); Helios Tsoi (hetsoi); Akash Deshpande (akdeshpa)
Cc: John Bettink (jbettink); Chris Satterlee (csatt); Raj Kalavendi (shekar); Siva Mortala (smortala); Raj Ammanur (rammanur); Kannan Varadhan (kvaradha); Pratibha Suryadevara (psuryade); Huy Ton (huyton); Mridul Bajpai (mridul); Siva Perumalla (sivap); Sudhir Rustogi (srustogi); Mahesh Chellappa (maheshc); Etienne Martineau (etmartin)
Subject: RE: Guest VM running 25% slower than host ?

Guys,

I think I finally got to the bottom of that hyper-threading mystery…

Basically both NUMA (multi-socket system) and Hyper-Threading brings a similar type of 'asymmetry' across CPUs on a given system:

- With NUMA, the CPUs located on a given socket will have optimal access speed to the DRAM connected on that same socket while 'remote' DRAM access will be much slower.

- With hyper-threading, siblings on the same physical CPU are sharing resources such as L1/L2 caches and the execution pipeline which may affect compute performance greatly.

Because of the shared pipeline, it is usually more efficient to schedule jobs on different physical core rather than on the same hyper-threading siblings (to avoid pipeline contention)

_ BUT at the same time _

Because of the shared L1/L2, it is usually more efficient to place a multi-threaded program on the same hyper-threading siblings (to minimize cache miss)

The above considerations are well taken care of by the HostOS scheduler but unfortunately the GuestOS scheduler doesn't have visibility over the pCPU/Sibling/vCPU layout required to make similar optimization. Similarly, with the kernel generation we are using, the same problem arise with NUMA such that the GuestOS doesn't have visibility over the pCPU/Socket/vCPU layout required to make optimal optimization.

In order to solve this problem:

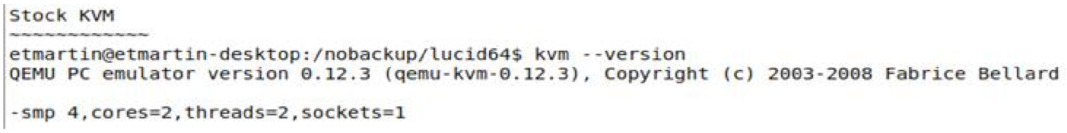
A) Make the Guest hyper-threading capable. For some reason this is broken on our thirdparty Qemu while this is well working on a older stock Qemu release. See **figure #1**

B) Using libvirt (or by hand with chrt) pin each vCPU to a pCPU while making sure the hyper-threading layout matches. Note that this is automatically taken care of by some newer libvirt implementation…

Now with that in mind (I did only 'B' because 'A' doesn't work on the router) I repeated the Dhrystone test on the Router / Calvados VM with everything up and running. The test is now _*consistently*_ showing the same performance than on the hostOS !

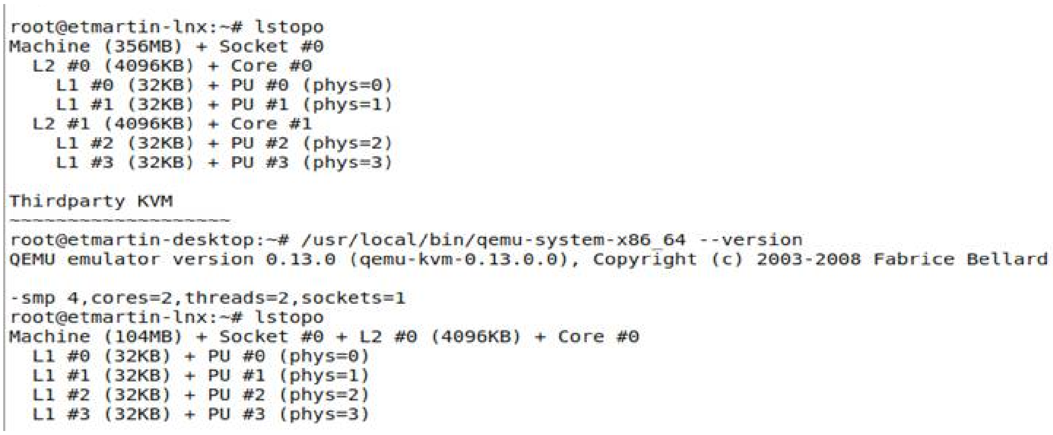
Figure #1

Show that with stock Qemu it is possible to create hyper-threaded vCPU but with thirdparty Qemu we cannot for some reason…



```
Stock KVM
~~~~~~~~~~~~
etmartin@etmartin-desktop:/nobackup/lucid64$ kvm --version
QEMU PC emulator version 0.12.3 (qemu-kvm-0.12.3), Copyright (c) 2003-2008 Fabrice Bellard

-smp 4,cores=2,threads=2,sockets=1
```

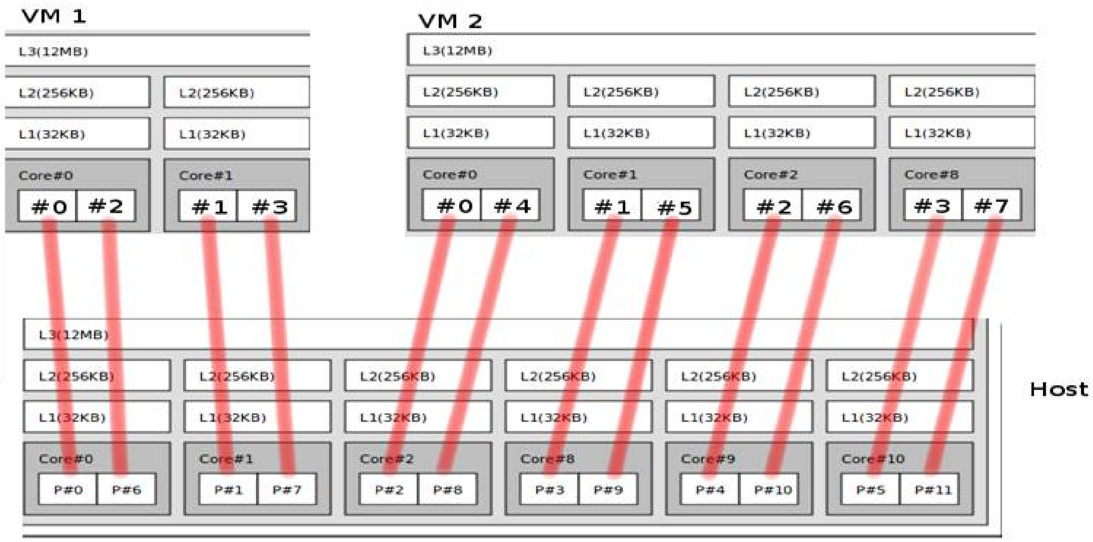


```
root@etmartin-lnx:~# lstopo
Machine (356MB) + Socket #0
  L2 #0 (4096KB) + Core #0
    L1 #0 (32KB) + PU #0 (phys=0)
    L1 #1 (32KB) + PU #1 (phys=1)
  L2 #1 (4096KB) + Core #1
    L1 #2 (32KB) + PU #2 (phys=2)
    L1 #3 (32KB) + PU #3 (phys=3)

Thirdparty KVM
~~~~~~~~~~~~~~~~~~~
root@etmartin-desktop:~# /usr/local/bin/qemu-system-x86_64 --version
QEMU emulator version 0.13.0 (qemu-kvm-0.13.0.0), Copyright (c) 2003-2008 Fabrice Bellard

-smp 4,cores=2,threads=2,sockets=1
root@etmartin-lnx:~# lstopo
Machine (104MB) + Socket #0 + L2 #0 (4096KB) + Core #0
  L1 #0 (32KB) + PU #0 (phys=0)
  L1 #1 (32KB) + PU #1 (phys=1)
  L1 #2 (32KB) + PU #2 (phys=2)
  L1 #3 (32KB) + PU #3 (phys=3)
```

This is the way the pCPU -> vCPU placement should be done when the GuestOS is Hyper threaded capable so that the guestOS scheduler SMT optimization kick in nicely.



Thanks,

Etienne

From: Etienne Martineau (etmartin)
Sent: Wednesday, May 22, 2013 7:04 PM
To: Etienne Martineau (etmartin); Venkat Garigipati (venkatg); Pankaj Bhagra (bhagra); Alec Hothan (ahothan); bajor(mailer list); Ajay Patel (ajaypat); Rajesh Ranga (raranga); Helios Tsoi (hetsoi); Akash Deshpande (akdeshpa)
Cc: John Bettink (jbettink); Chris Satterlee (csatt); Raj Kalavendi (shekar); Siva Mortala (smortala); Raj Ammanur (rammanur); Kannan Varadhan (kvaradha); Pratibha Suryadevara (psuryade); Huy Ton (huyton); Mridul Bajpai (mridul); Siva Perumalla (sivap); Sudhir Rustogi (srustogi); Mahesh Chellappa (maheshc); Etienne Martineau (etmartin)
Subject: RE: Guest VM running 25% slower than host ?

Gents,

Part of the KVM real time investigation work that I'm doing I found something that may very well explain why we are seeing performance degradation in the VM (poor Niantic performance / low & inconsistent Dhrystone). Let me explain:

The Sandybridge CPU is configured to run in hyper-threaded mode. With hyper-threading there is two register set per physical CPU but only _one_ execution pipeline which is responsible to ‘crunch’ the opcode stream. That pipeline runs at the CPU core frequency hence if we have two CPU intensive task running over a common hyper-threaded CPU they will effectively run at 50% of the nominal CPU capacity.

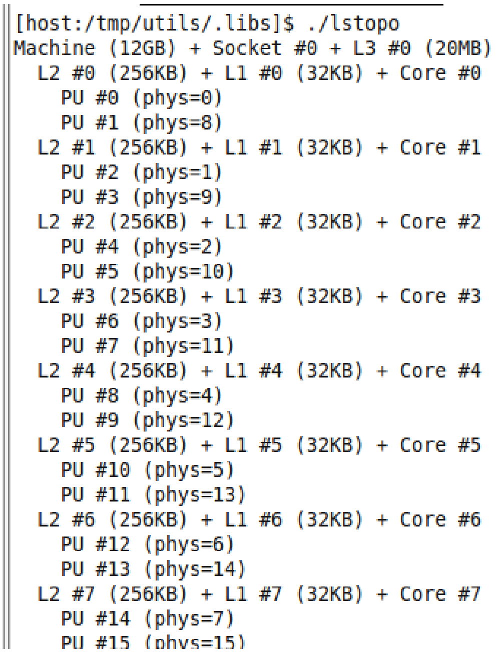
The **figure #1** below illustrate the geographical location of the hyper-threaded CPUs with respect to physical core. For example, the hyper threaded pair CPU 0 / CPU 8 is located on Core 0. Similarly, the hyper-threaded pair CPU 1 / CPU 9 is located on Core 1 and so on.

So as a first step in order to verify my theory I launched the Calvados VM and assign vCPU0 to pCPU0 and vCPU1 to pCPU1 so that _both_ vCPU in Calvados are _not_ located on a common hyper-threaded pair. I launched my test program on both vCPU simultaneously. This test program measure very precisely the amount of CPU cycle it takes to execute a given loop n times. **Figure #2** illustrate the results. We can observe that in average it takes 100000 cycle to execute the loop. Note that the ripples are caused by external Interruptions in both guest Kernel and host Kernel.

As a second step, I launched the Calvados VM and assign vCPU0 to pCPU0 and vCPU1 to pCPU8 so that _both_ vCPU in Calvados are always running on a hyper-threaded pair located on the Core 0. **Figure #3** illustrate the results. We can clearly observe that it takes much more CPU time to execute the given loop n times.

So all in all, hyper-threading brings additional throughput to the system by cutting in half the number of context switch at the expense of a lack of determinism in raw CPU computation power.

Figure #1



```
[host:/tmp/utils/.libs]$ ./lstopo
Machine (12GB) + Socket #0 + L3 #0 (20MB)
  L2 #0 (256KB) + L1 #0 (32KB) + Core #0
    PU #0 (phys=0)
    PU #1 (phys=8)
  L2 #1 (256KB) + L1 #1 (32KB) + Core #1
    PU #2 (phys=1)
    PU #3 (phys=9)
  L2 #2 (256KB) + L1 #2 (32KB) + Core #2
    PU #4 (phys=2)
    PU #5 (phys=10)
  L2 #3 (256KB) + L1 #3 (32KB) + Core #3
    PU #6 (phys=3)
    PU #7 (phys=11)
  L2 #4 (256KB) + L1 #4 (32KB) + Core #4
    PU #8 (phys=4)
    PU #9 (phys=12)
  L2 #5 (256KB) + L1 #5 (32KB) + Core #5
    PU #10 (phys=5)
    PU #11 (phys=13)
  L2 #6 (256KB) + L1 #6 (32KB) + Core #6
    PU #12 (phys=6)
    PU #13 (phys=14)
  L2 #7 (256KB) + L1 #7 (32KB) + Core #7
    PU #14 (phys=7)
    PU #15 (phys=15)
```

Figure #2

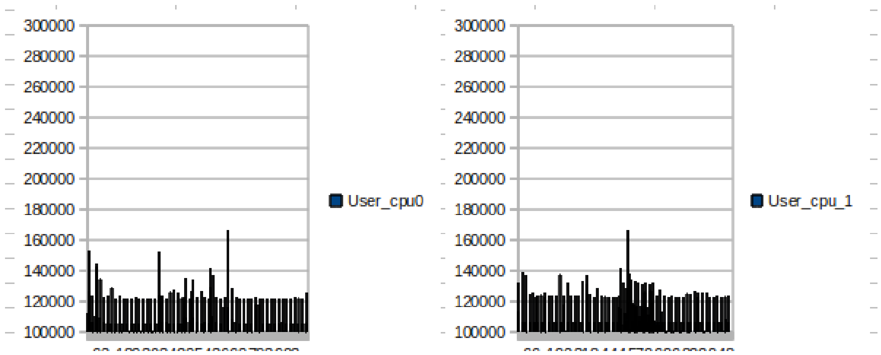
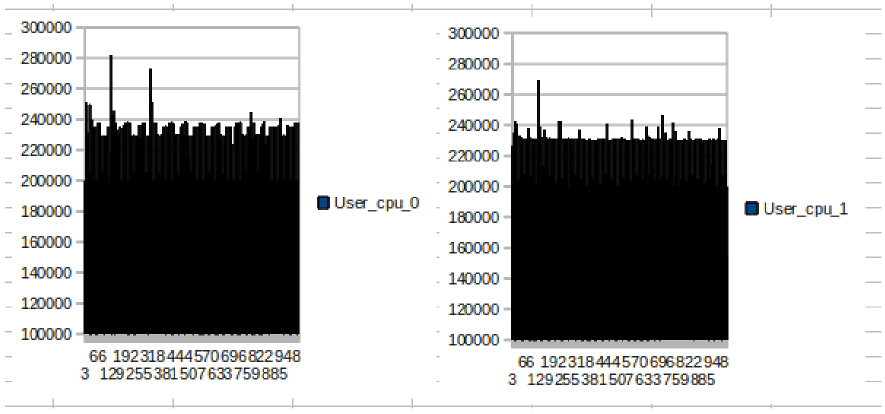


Figure #3



Thanks,
Etienne

From: Etienne Martineau (etmartin)
Sent: Monday, May 13, 2013 3:11 PM
To: Venkat Garigipati (venkatg); Pankaj Bhagra (bhagra); Alec Hothan (ahothan); bajor(mailer list); Ajay Patel (ajaypat); Rajesh Ranga (raranga); Helios Tsoi (hetsoi)
Cc: John Bettink (jbettink); Chris Satterlee (csatt); Raj Kalavendi (shekar); Siva Mortala (smortala); Raj Ammanur (rammanur); Akash Deshpande (akdeshpa); Kannan Varadhan (kvaradha); Pratibha Suryadevara (psuryade); Huy Ton (huyton); Mridul Bajpai (mridul); Siva Perumalla (sivap); Sudhir Rustogi (srustogi); Mahesh Chellappa (maheshc)
Subject: RE: Guest VM running 25% slower than host ?

Gents,

Out of curiosity I ran my own benchmark tool which consist of a special test program that has been designed to measure _very_ precisely the amount of CPU cycle it takes to execute a given piece of code (bus cycle accurate). The test program is designed so that is can be executed in both kernel space and user space.

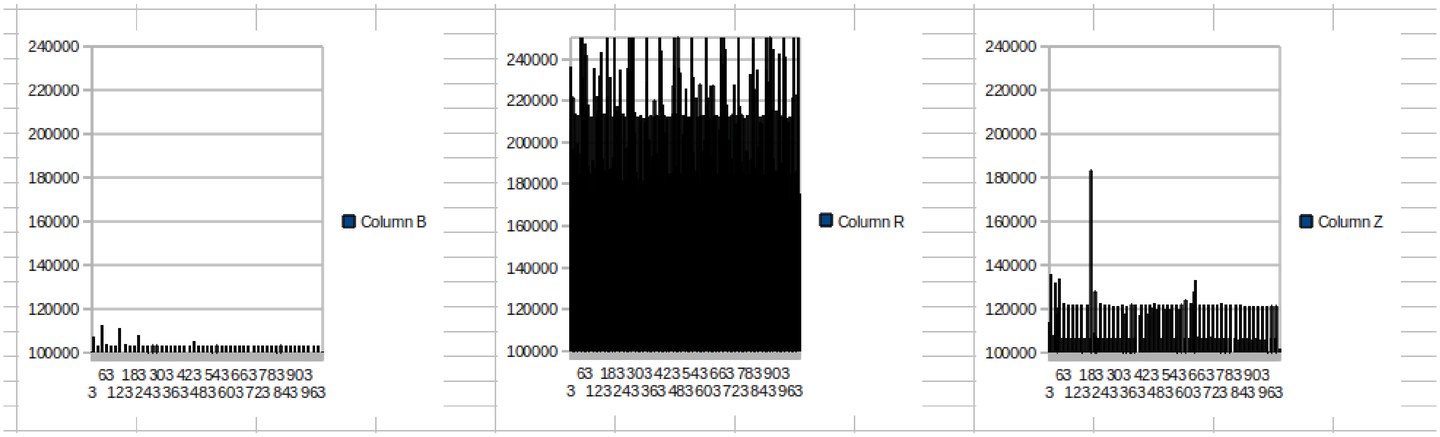
The left graph below illustrate the results when the test runs on HostOS user space. In average the number of CPU cycles needed to execute the test code is 100000 but as you can see there is some little ripple at regular interval. This is when the kernel timer IRQ handler is executing on the same CPU where the test is executing (remember that user space can be interrupted by kernel space IRQ…)

The middle graph illustrate the results when the test runs on GuestOS (CalvadosVM) user space. As we can see there is a _*lots*_ of noise and the performance degradation is inline with previous observation from other folks from this email thread.

After some more investigation I found that the test loop was constantly interrupted (at high freq) by other processes running in the VM. I also noticed that the other processes were not performing much work other that presumably handling some event and yielding the CPU right away.

So I decided to bump the test program in RR 80 real time class to avoid the noise from those other processes. The right graph illustrate the results in that case. The performance degradation (when compared to hostOS user space) is 1.54% which is inline with the ‘cost of virtualization’ typically advertise by the KVM community.

So all in all from a relative standpoint (nothing absolute WRT to sandybridge) I don’t see any problem related to KVM / VMexit or Kernel. There is obviously some application that are flooding the scheduler’s run-queue but this is another problem all together. NOTE: I haven’t tried on XR VM but I suspect similar behavior…



Thanks,

Etienne

From: Venkat Garigipati (venkatg)
Sent: Monday, May 13, 2013 12:26 PM
To: Pankaj Bhagra (bhagra); Alec Hothan (ahothan); bajor(mailer list); Etienne Martineau (etmartin); Ajay Patel (ajaypat); Rajesh Ranga (raranga); Helios Tsoi (hetsoi)
Cc: John Bettink (jbettink); Chris Satterlee (csatt); Raj Kalavendi (shekar); Siva Mortala (smortala); Raj Ammanur (rammanur); Akash Deshpande (akdeshpa); Kannan Varadhan (kvaradha); Pratibha Suryadevara (psuryade); Huy Ton (huyton); Mridul Bajpai (mridul); Siva Perumalla (sivap); Sudhir Rustogi (srustogi)
Subject: Re: Guest VM running 25% slower than host ?

No, I am not saying its an NTP issue, but wanted to rule that factor out. I don't think we should be having this issue as well. Is the CPU clock tuned to be in sync ?. Please let us know if this is an issue.

Thanks//Venkat

From: "Pankaj Bhagra (bhagra)" <bhagra@cisco.com>
Date: Sunday, May 12, 2013 12:23 AM
To: Cisco Employee <venkatg@cisco.com>, "Alec Hothan (ahothan)" <ahothan@cisco.com>, "bajor(mailer list)" <bajor@cisco.com>, "Etienne Martineau (etmartin)" <etmartin@cisco.com>, "Ajay Patel (ajaypat)" <ajaypat@cisco.com>, "Rajesh Ranga (raranga)" <raranga@cisco.com>, "Helios Tsoi (hetsoi)" <hetsoi@cisco.com>
Cc: "John Bettink (jbettink)" <jbettink@cisco.com>, "Chris Satterlee (csatt)" <csatt@cisco.com>, "Raj Kalavendi (shekar)" <shekar@cisco.com>, "Siva Mortala (smortala)" <smortala@cisco.com>, "Raj Ammanur (rammanur)" <rammanur@cisco.com>, "Akash Deshpande (akdeshpa)" <akdeshpa@cisco.com>, "Kannan Varadhan (kvaradha)" <kvaradha@cisco.com>, "Pratibha Suryadevara (psuryade)" <psuryade@cisco.com>, "Huy Ton (huyton)" <huyton@cisco.com>, "Mridul Bajpai (mridul)" <mridul@cisco.com>, "Siva Perumalla (sivap)" <sivap@cisco.com>, "Sudhir Rustogi (srustogi)" <srustogi@cisco.com>
Subject: RE: Guest VM running 25% slower than host ?

Its an issue which need to be looked. But I am not sure how u r mapping this to NTP drift or how it would be related to it?

From: Venkat Garigipati (venkatg)
Sent: Saturday, May 11, 2013 10:52 PM
To: Pankaj Bhagra (bhagra); Alec Hothan (ahothan); bajor(mailer list); Etienne Martineau (etmartin); Ajay Patel (ajaypat); Rajesh Ranga (raranga); Helios Tsoi (hetsoi)

Cc: John Bettink (jbettink); Chris Satterlee (csatt); Raj Kalavendi (shekar); Siva Mortala (smortala); Raj Ammanur (rammanur); Akash Deshpande (akdeshpa); Kannan Varadhan (kvaradha); Pratibha Suryadevara (psuryade); Huy Ton (huyton); Mridul Bajpai (mridul); Siva Perumalla (sivap); Sudhir Rustogi (srustogi)
Subject: Re: Guest VM running 25% slower than host ?

Hi folks,

So is this possibly due to the NTP drift or the out of sync issue ?. Is there a closure or update on this issue ?. This could be an impact to the control plane/forwarding plane scale and performance if its really an issue.

Thanks//Venkat

From: "Pankaj Bhagra (bhagra)" <bhagra@cisco.com>
Date: Friday, May 10, 2013 6:03 PM
To: "Alec Hothan (ahothan)" <ahothan@cisco.com>, "bajor(mailer list)" <bajor@cisco.com>, "Etienne Martineau (etmartin)" <etmartin@cisco.com>, "Ajay Patel (ajaypat)" <ajaypat@cisco.com>, "Rajesh Ranga (raranga)" <raranga@cisco.com>, "Helios Tsoi (hetsoi)" <hetsoi@cisco.com>
Cc: "John Bettink (jbettink)" <jbettink@cisco.com>, "Chris Satterlee (csatt)" <csatt@cisco.com>, "Raj Kalavendi (shekar)" <shekar@cisco.com>, "Siva Mortala (smortala)" <smortala@cisco.com>, "Raj Ammanur (rammanur)" <rammanur@cisco.com>, "Akash Deshpande (akdeshpa)" <akdeshpa@cisco.com>, "Kannan Varadhan (kvaradha)" <kvaradha@cisco.com>, "Pratibha Suryadevara (psuryade)" <psuryade@cisco.com>, "Huy Ton (huyton)" <huyton@cisco.com>, "Mridul Bajpai (mridul)" <mridul@cisco.com>, "Siva Perumalla (sivap)" <sivap@cisco.com>, "Sudhir Rustogi (srustogi)" <srustogi@cisco.com>
Subject: RE: Guest VM running 25% slower than host ?

Hi Alec,

The test ran for 200sec or so (1B rounds).

I didn't tune the cpu clock.

Only thing I parameterize:

- Optimization level to O0.
- Memory clock speed to 1600Mhz to match the RP/LC mem speed.

What do I need to tune the cpu clock ?

This is the ptr to makefile /ws/bhagra-sjc/Makefile

Let me know if u see any other config to be tuned. thanks.

Later,

pankaj

From: Alec Hothan (ahothan)
Sent: Friday, May 10, 2013 7:50 AM
To: Pankaj Bhagra (bhagra); bajor(mailer list); Etienne Martineau (etmartin); Ajay Patel (ajaypat); Rajesh Ranga (raranga); Helios Tsoi (hetsoi)
Cc: John Bettink (jbettink); Chris Satterlee (csatt); Raj Kalavendi (shekar); Siva Mortala (smortala); Raj Ammanur (rammanur); Akash Deshpande (akdeshpa); Kannan Varadhan (kvaradha); Pratibha Suryadevara (psuryade); Huy Ton (huyton); Mridul Bajpai (mridul); Siva Perumalla (sivap); Sudhir Rustogi (srustogi)
Subject: Re: Guest VM running 25% slower than host ?

Hi Pankaj

I'd suggest that you run the test in a loop a large number of times, the total time should be several seconds.

Can you also describe how do you tune the cpu clock?

Thanks

Alec

From: "Pankaj Bhagra (bhagra)" <bhagra@cisco.com>
Date: Friday, May 10, 2013 12:31 AM
To: "bajor(mailer list)" <bajor@cisco.com>, "Etienne Martineau (etmartin)" <etmartin@cisco.com>, "Ajay Patel (ajaypat)" <ajaypat@cisco.com>, "Rajesh Ranga (raranga)" <raranga@cisco.com>, "Helios Tsoi (hetsoi)" <hetsoi@cisco.com>
Cc: "John Bettink (jbettink)" <jbettink@cisco.com>, "Chris Satterlee (csatt)" <csatt@cisco.com>, Alec Hothan <ahothan@cisco.com>, "Raj Kalavendi (shekar)" <shekar@cisco.com>, "Siva Mortala (smortala)" <smortala@cisco.com>, "Raj Ammanur (rammanur)" <rammanur@cisco.com>, "Akash Deshpande (akdeshpa)" <akdeshpa@cisco.com>, "Kannan Varadhan (kvaradha)" <kvaradha@cisco.com>, "Pratibha Suryadevara (psuryade)" <psuryade@cisco.com>, "Huy Ton (huyton)" <huyton@cisco.com>, "Mridul Bajpai (mridul)" <mridul@cisco.com>, "Siva Perumalla (sivap)" <sivap@cisco.com>
Subject: Guest VM running 25% slower than host ?

> Guys,
>
> I ran the Dhrystone test (complied with –O 0) on both guest and host (multiple iteration across multiple hw testbed – similar result). The build is done for native 2.6.34 and fired on the host and guest kernel. While the test was running their was no excessive interrupts, io or cpu load was offered.
>
> http://zenit.senecac.on.ca/wiki/index.php/Dhrystone_howto
>
> Two observations
>
> a) RP host is pegged at lower DMIPS (2190) irrespective of the cpu core clock. Tried on two cpu with cpu speed at 1400Mhz and 1800Mhz. Thus the guest also is also pegged at the same DMIPS. No difference in behavior on host or guest. But I think it's bcos the host itself is getting slow here for some reason. Same behavior across testbeds.
>
> b) LC host runs average ~25% faster than LC guest (both calvados and XR). DMIPS 3557/2745 – average of multiple runs across various speed LC (2000Mhz and 1400Mhz).
>
> Can someone from the OS team evaluate this more carefully. We need to look at this issue with priority. Thanks.
>
> /ws/bhagra-sjc/gcc_dry2
>
> Later,
>
> Pankaj
>
> // core clock 1400Mhz and 1800Mhz
>
> In host of RP
>
> Microseconds for one run through Dhrystone: 0.3
>
> Dhrystones per Second: 3846153.8
>
> In Guest of RP
>
> Microseconds for one run through Dhrystone: 0.3
>
> Dhrystones per Second: 3846153.8
>
> // Core clock 2000Mhz

> On LC calvados VM
>
> ```
> Microseconds for one run through Dhrystone:    0.2
> Dhrystones per Second:                      5555555.5 (to 4545454.5)
> ```
>
> On LC host
>
> ```
> Microseconds for one run through Dhrystone:    0.2
> Dhrystones per Second:                      6250000.0
> ```
>
> // Core clock 1400Mhz
>
> On LC host
>
> ```
> Microseconds for one run through Dhrystone:    0.2
> Dhrystones per Second:                      4347826.0
> ```
>
> On LC guest
>
> ```
> Microseconds for one run through Dhrystone:    0.3
> Dhrystones per Second:                      3303703.8
> ```